JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS 124, 33–42 (1987)

A Shortest Path Routing Problem with Resource Allocation

G. PEDERZOLI

Concordia University, Montreal, Canada

AND

N. G. F. SANCHO

McGill University, Montreal, Canada

Submitted by E. Stanley Lee

Received August 1, 1985

This paper considers a variation of the short path problem in which the time required to traverse an arc is a function of the discrete amount of resource used up while traversing the arc. A short path is sought subject to a limit on the amount of resource available. Two algorithms are proposed for the solutions of this problem. A rather strict monotonicity assumptions ensures that all available resources will be used. © 1987 Academic Press, Inc.

1. INTRODUCTION

In the original shortest path routing problem a set of points numbered $1, 2,..., N$ is given with N representing the terminal, and it is assumed that there exists a direct link between any two points i and j. The time required to traverse the arc from i to j will be denoted by t_{ij}, with $t_{ii}=0$ for $i=1, 2,..., N$. In what follows $t_{ij} \geqslant 0$ if $i \neq j$ and $t_{ij}=\infty$ if there is no direct link from point i to point j. The problem of finding the shortest path from any point to the terminal has been considered by many authors, see, for instance, Boffey [2].

The dynamic programming approach suggested by Dreyfus and Law [4] is formulated in the following way. Let

$$f_i = \text{minimum time required to go from } i \text{ to } N \tag{1}$$

where $i=1, 2,..., N-1$ and the set $f_N=0$. Then the principle of optimality yields the recursive relation

$$f_i = \min_{i \neq j} [t_{ij} + f_j], \qquad i=1, 2,..., N-1 \tag{2}$$

0022-247X/87 $3.00

with the boundary condition $f_N = 0$, and it is shown in Bellman [1] that the recursion (2) possesses a unique solution. A computational algorithm is proposed which converges to the functional equation (2).

2. The Routing Problem with Resource Allocation

We now consider the same shortest path routing problem as described in the Introduction. However, associated with the time to go from point i to point j, we assume a certain amount of resource is used up. Hence we define $t_{ij}(m)$ as the time to go from i to j using up an amount m of resources in the process. It is also assumed that $t_{ij}(0) = \infty$, i.e., if no resource is used up then it is impossible to get from i to j. This assumption will be relaxed in another section when we consider the allocation of zero resources. We may also stipulate that the more resources are used up the smaller will be the time to go from i to j. Hence we may write

$$t_{ij}(m_1) < t_{ij}(m_2) \qquad \text{if} \quad m_1 > m_2. \tag{3}$$

The problem to be considered can now be described as follows. We are required to get from point 1 to N in minimum time when a total of n resources are available. Let

$$f_i(n) = \text{the minimum time to go from } i \text{ to } N \text{ using up an amount of } n \text{ units of resources in the process} \tag{4}$$

with the boundary condition

$$f_N(0) = 0, \qquad \text{i.e., all the resources are used up when we arrive at point } N. \tag{5}$$

The principle of optimality then yields the recursive relation

$$f_i(n) = \min_{\substack{m \\ j \neq i}} [t_{ij}(m) + f_j(n-m)] \tag{6}$$

where $n = 1, 2, 3,...$ and for any integer m such that $1 \leqslant m \leqslant n$. The minimum on the right-hand side of (6) is given with respect to m and $j \neq i$. Recursion (6) is a variation on a method described by Lawler [6] where multiple arcs between nodes are used to model time-resource usage pairs.

The algorithm for solving the recursive equation in (6) can be found as follows:

(i) Compute $f_i(1)$ for all appropriate i's. This gives all direct paths from i to N using one unit of resource.

(ii) Next compute $f_i(2)$ for all appropriate i's using $f_i(1)$.

(iii) Next $f_i(3)$ for all appropriate i's using $f_i(1)$ and $f_i(2)$.

(iv) Continue until $f_i(n)$ is computed where n is total initial resource.

A numerical illustration is given in another section. The problem is somewhat more complicated if allocations of zero are allowed. In this case the above algorithm is not very practical since in order to find $f_i(0)$, $f_i(1)$, etc., one must also know $f_j(0)$, $f_j(1)$, etc., respectively, but in (6), $f_j(n-m)$ is not known explicitly. Hence another algorithm is proposed if zero allocations are allowed.

3. Approximation in Policy Space

Let us define a function $i(j, m)$ to be a policy which tells us the point j to go from i using an amount m of resources. A path solution can then be given by $[i, j_1, j_2, ..., j_{k-1}, N]$, where $i = i(j_1, m_1)$, $j_1 = j_1(j_2, m_2), ..., j_{k-1} = j_{k-1}(N, m_k)$ and such that $m_1 + m_2 + \cdots + m_k = n$, i.e., when a total of n resources are used up in the process.

Assume now a policy $i(j, m)$ which enables us to go from i to N using up a total of n resources. Let $f_i^{(0)}(n)$ be calculated using this policy, i.e.,

$$\begin{aligned} f_i^{(0)}(n) &= t_{ij_0}(m) + f_{j_0}^{(0)}(n-m), \qquad n = 0, 1, 2, 3, ... \text{ and } 0 \leqslant m \leqslant n \\ &= t_{ij_0}(m) + \cdots \end{aligned} \tag{7}$$

where the dots indicate the terms obtained by iterating the relation. This is now used as an initial approximation which can be used as a starting point in the method described below.

With $f_i^{(0)}(n)$ as given in (7) we determine j and m to minimize the expression $t_{ij}(m) + f_j^{(0)}(n-m)$, which we call j_1 and $\bar{m}$, respectively. This can be written as

$$f_i^{(1)}(n) = t_{ij_1}(\bar{m}) + f_{j_1}^{(0)}(n-\bar{m}). \tag{8}$$

Thus we may assert that

$$t_{ij_1}(\bar{m}) + f_{j_1}^{(0)}(n-\bar{m}) \leqslant t_{ij_0}(m) + f_{j_0}^{(0)}(n-m) \tag{9}$$

i.e.,

$$f_i^{(1)}(n) \leqslant f_i^{(0)}(n) \qquad i\text{'s, and } n = 0, 1, 2, ... \tag{10}$$

Continuing on with this iteration for all i's we obtain

$$f_i^{(0)}(n) \geqslant f_i^{(1)}(n) \geqslant \cdots \geqslant f_i^{(r)}(n) \geqslant \cdots \geqslant f_i(n) \tag{11}$$

where in general we can write

$$f_i^{(r)}(n) = \min_{\substack{m \\ j \neq i}} [t_{ij}(m) + f_j^{(r-1)}(n-m)], \qquad r = 1, 2, 3, \ldots \tag{12}$$

If $f_i^{(r)}(n) = f_i^{(r+1)}(n) = \bar{f}_i(n)$, say for all i's and n's then the algorithm has converged since we have for $\bar{f}_i(n)$ the equation

$$\bar{f}_i(n) = \min_{\substack{m \\ j \neq i}} [t_{ij}(m) + \bar{f}_j(n-m)] \tag{13}$$

which is the same equation for $f_i(n)$ given in (6). Uniqueness will show that $\bar{f}_i(n) = f_i(n)$.

The advantage of this method is that if a good approximation in policy space is initially chosen then the method can be very efficient. It can be shown that $f_i^{(r)}(n)$ converges within $N-1$ moves. A proof of this when $n=0$ is given by Davidson and White [3], but the result holds for any value of n, since we are minimizing with respect to m and j instead of j only.

This problem is also solvable by other methods, such as the vector optimization version of the standard Bellman–Ford technique described in Hartley [5].

4. Existence and Uniqueness

The existence of a solution for the functional equation given in (6) is immediate. There is a shortest path from i to N using up n resources, since there are only a finite number of admissible paths, namely those containing no loops. The time required to traverse this path (which need not be unique) defines a function $g_i(n)$, $i = 1, 2, \ldots, N-1$ with $g_N(0) = 0$.

A path of minimum length must go to some other point k using up an amount of resources m. Hence we have

$$g_i(n) = t_{ik}(m) + g_k(n-m) \tag{14}$$

for some k and m. This value of k and m must minimize the right-hand side, since it would contradict the definition of $g_i(n)$. Hence

$$g_i(n) = \min_{\substack{m \\ k \neq i}} [t_{ik}(m) + g_k(n-m)], \tag{15}$$

i.e., $g_i(n)$ constitutes a nonnegative solution of Eq. (6).

For a proof of uniqueness we assume that there are two distinct solutions $g_i(n)$ and $h_i(n)$ for $i=1, 2, \ldots, N$ and $n=0, 1, 2, \ldots$ with boundary condition $g_N(0)=h_N(0)=0$. We also assume that the more resources are used up from point i, the smaller will be the total travel time. Hence we have

$$g_i(n_1)>g_i(n_2) \text{ if } n_2>n_1 \qquad \text{and} \qquad h_i(n_1)>h_i(n_2) \text{ if } n_2>n_1. \tag{16}$$

Suppose that for point 1 we have, by renaming $g_1(n)$ and $h_1(n)$ if necessary, that $g_1(n)-h_1(n)>0$. On the other hand, if they are identical then $g_1(n)-h_1(n)=0$. Thus we can write

$$g_1(n)-h_1(n)\geqslant 0 \tag{17}$$

and it is easy to see that

$$g_1(n)=\min_{\substack{m\\ j\neq i}} [t_{1j}(m)+g_j(n-m)] \tag{18}$$

and

$$h_1(n)=\min_{\substack{m\\ j\neq i}} [t_{1j}(m)+h_j(n-m)]. \tag{19}$$

Suppose the values of m and j giving the minimum in (19) are $m=m_1$ and $j=2$, respectively. This value for $j=2$ can always be arranged by renumbering the vertices 2 to N if necessary. It follows that

$$g_1(n)=t_{1j}(m_1)+g_j(n-m_1) \qquad \text{for} \quad j=2, 3, \ldots, N; 0\leqslant n_1\leqslant n \tag{20}$$

and

$$h_1(n)=t_{12}(m_1)+h_2(n-m_1). \tag{21}$$

This relation leads to

$$\begin{aligned} g_1(n)-h_1(n)&\leqslant t_{12}(m_1)+g_2(n-m_1)-h_1(n)\\ &=g_2(n-m_1)-h_2(n-m_1) \end{aligned}$$

i.e.,

$$g_1(h)-h_1(n)\leqslant g_2(n-m_1)-h_2(n-m_1). \tag{22}$$

Continuing on we can similarly write as in (18) and (19),

$$g_2(n-m_1)=\min_{\substack{m\\ j\neq 2}} [t_{2j}(m)+g_j(n-m_1-m)], \qquad 0\leqslant m\leqslant n-m_1 \tag{23}$$

and

$$h_2(n-m_1)=\min_{\substack{m\\ j\neq 2}}\,[t_{2j}(m)+h_j(n-m_1-m)]. \tag{24}$$

The minimum value of j in (24) cannot be 1 with $m=m_2$, since

$$t_{21}(m_2)+h_1(n-m_1-m_2)>t_{21}(m_2)+h_1(n)$$

and by assumption $h_1(n_1)>h_1(n_2)$ if $n_2>n_1$. Therefore

$$t_{21}(m_2)+h_1(n-m_1-m_2)>t_{21}(m_2)+t_{12}(m_1)+h_2(n-m_1)>h_2(n-m_1). \tag{25}$$

If the values of m and j giving the minimum in (24) are $m=m_2$ and $j=3$, which we can arrange by renumbering the vertices 3 to N if necessary, we then obtain

$$g_2(n-m_1)\leqslant t_{2j}(m_2)+g_j(n-m_1-m_2), \qquad j=1,3,4,\ldots,N \tag{26}$$

and

$$h_2(n-m_1)=t_{23}(m_2)+h_3(n-m_1-m_2). \tag{27}$$

Hence

$$\begin{aligned} g_2(n-m_1)-h_2(n-m_1) &\leqslant t_{23}(m_2)+g_3(n-m_1-m_2)-h_2(n-m_1)\\ &= g_3(n-m_1-m_2)-h_3(n-m_1-m_2). \end{aligned} \tag{28}$$

Therefore the inequality in (22) can now be written as

$$g_1(n)-h_1(n)\leqslant g_3(n-m_1-m_2)-h_3(n-m_1-m_2). \tag{29}$$

Continuing on as before we finally arrive at

$$g_1(n)-h_1(n)\leqslant g_N(0)-h_N(0) \tag{30}$$

where we assume that all our resources have been used up when we have reached the final destination N. Note, however, that the boundary condition gives $g_N(0)=h_N(0)=0$ and therefore can be written as

$$g_1(n)-h_1(n)\leqslant 0. \tag{31}$$

Comparing (31) with (17) we obtain $g_1(n)-h_1(n)=0$.

Repeating the same argument for the next point 2 and renaming $g_2(n)$

and $h_2(n)$ if necessary, we have that $g_2(n) - h_2(n) > 0$. If $g_2(n)$ and $h_2(n)$ are identical, then of course $g_2(n) - h_2(n) = 0$, and therefore we can write

$$g_2(n) - h_2(n) \geqslant 0. \tag{32}$$

Following the same procedure described above we have

$$g_2(n) - h_2(n) \leqslant 0 \tag{33}$$

and comparing (32) with (33) we conclude that $g_2(n) - h_2(n) = 0$. Continuing on in this manner we obtain $g_3(n) - h_3(n) = 0, \ldots,$ and $g_{N-1}(n) - h_{N-1}(n) = 0$, thus proving that the two solutions are identical.

5. Numerical Illustration

In this example we are required to go from point 1 to 6 using an amount of $n = 8$ resources in the process. A graph of the possible paths is shown in Fig. 1. Furthermore, it is assumed that at least one unit of resource must be spent in going from point i to point j. Using the first algorithm described in Section 2 we first compute the values of $f_i(1)$. If a value of ∞ is obtained it means that there is no way to go from 1 to 6 using one unit of resource. Next, using $f_i(1)$ from the recursion (6) we obtain $f_i(2)$ and so on. The values of times $t_{ij}(m)$ to go from i to j using up an amount m of resources are given in Table I. The results obtained for the first algorithm are given in Table II. The term in brackets (j, m) gives the point j to go to from i and the amount m of resources used up in the process.

We see from Table II that the value for $f_1(8) = 13.0$ (2, 3), i.e., from point 1 we go to point 2 using 3 units of resources giving the time $t_{12}(3)$ in the process. Next we read off the value $f_2(5) = 9$ (4, 4), i.e., from point 2 we go to point 4 using 4 units of resources and giving the time $t_{24}(4)$ in the process. Finally we read off the value $f_4(1) = 8$ (6, 1), i.e., from point 4 wego to point 6 using 1 unit of resource and again giving the time $t_{46}(1)$ in the process. A check of Table I shows that $t_{12}(3) + t_{24}(4) + t_{46}(1) = 13$.

We next assume that zero allocations are allowed with the following additional values given

$t_{12}(0)$	$t_{13}(0)$	$t_{23}(0)$	$t_{24}(0)$	$t_{25}(0)$	$t_{32}(0)$	$t_{34}(0)$	$t_{35}(0)$	$t_{45}(0)$	$t_{46}(0)$	$t_{54}(0)$	$t_{56}(0)$
15	15	10	25	12	8	7	10	15	10	15	10

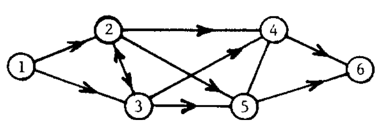

Figure 1.

TABLE I

$t_{ij}(m)$ \ (m)	1	2	3	4	5	6	7	8
$t_{12}(m)$	10.0	6.0	4.0	3.0	—	—	—	—
$t_{13}(m)$	9.0	6.5	4.5	3.5	3.0	—	—	—
$t_{23}(m)$	6.0	5.0	4.0	3.0	2.0	1.4	1.0	0.7
$t_{24}(m)$	20.0	12.0	5.0	1.0	0.8	0.7	0.65	0.64
$t_{25}(m)$	8.0	7.0	6.1	5.3	4.6	4.0	3.5	3.1
$t_{32}(m)$	6.0	5.0	4.0	3.0	2.0	1.4	1.0	0.7
$t_{34}(m)$	5.0	4.0	3.2	2.5	1.9	1.4	1.1	0.9
$t_{35}(m)$	7.0	5.5	4.3	3.3	2.4	2.2	2.1	2.0
$t_{45}(m)$	12.0	8.0	5.0	3.0	2.0	1.5	1.1	1.0
$t_{46}(m)$	8.0	7.4	6.9	6.5	6.2	6.0	5.8	5.7
$t_{54}(m)$	12.0	8.0	5.0	3.0	2.0	1.5	1.1	1.0
$t_{56}(m)$	7.5	7.0	6.5	6.0	5.5	5.0	4.5	4.0

We are again required to go from point 1 to point 6 using up 6 units of resources. As a first approximation in policy space we can assume the path 1—2—4—6 from point 1 or the path 3—5—6 from point 3, etc., using up an amount of resources 0, 1, 2,..., 8 from point 1 or 0, 1, 2,..., 8 from point 3, etc. By choosing the particular policy for the resource to be used up we can arrive at $f_i^{(0)}(n)$ and to be realistic we always choose $f_i^{(0)}(n)$ such that

TABLE II

$f_i(n)$ \ (n)	1	2	3	4	5	6	7	8
$f_1(n)$	∞	∞	22.0 (3, 1)	19.5 (3, 2)	17.5 (3, 3)	16.5 (3, 3) (3, 4)	15.0 (2, 2)	13.0 (2,3)
$f_2(n)$	∞	15.5 (5, 1)	14.5 (5, 2)	13.0 (4, 3)	9.0 (4, 4)	8.4 (4, 4)	7.9 (4, 4)	7.5 (4, 4)
$f_3(n)$	∞	13.0 (4, 1)	12.0 (4, 2)	11.20 (4, 3)	10.5 (4, 4)	9.9 (4, 5) (5, 5)	9.4 (4, 6) (5, 5)	8.8 (4, 5) (4, 6)
$f_4(n)$	8.0 (6, 1)	7.4 (6, 2)	6.9 (6, 3)	6.5 (6, 4)	6.2 (6, 5)	6.0 (6, 6)	5.8 (6, 7)	5.7 (6, 8)
$f_5(n)$	7.1 (6, 1)	7.0 (6, 2)	6.5 (6, 3)	6.0 (6, 4)	5.5 (6, 5)	5.0 (6, 6)	4.5 (6, 7)	4.0 (6, 8)

TABLE III

(n) / $f_i^{(0)}(n)$	0	1	2	3	4	5	6	7	8
$f_1^{(0)}(n)$	50 (2, 0)	48 (2, 0)	43 (2, 0)	38.0 (2, 1)	30.0 (2, 1)	29.4 (2, 1)	22.4 (2, 1)	18.4 (2,1)	17.9 (2, 1)
$f_2^{(0)}(n)$	35 (4, 0)	33 (4, 0)	28 (4, 1)	20.0 (4, 2)	19.4 (4, 2)	12.4 (4, 3)	8.4 (4, 4)	7.9 (4, 4)	7.5 (4, 4)
$f_3^{(0)}(n)$	20 (5, 0)	17.5 (5, 0)	14.5 (5, 1)	13.0 (5, 2)	12.5 (5, 2)	11.3 (5, 3)	10.8 (5, 3)	9.8 (5, 4)	9.3 (5, 4)
$f_4^{(0)}(n)$	10 (6, 0)	8.0 (6, 1)	7.4 (6, 2)	6.9 (6, 3)	6.5 (6, 4)	6.2 (6, 5)	6.0 (6, 6)	5.8 (6, 7)	5.7 (6, 8)
$f_5^{(0)}(n)$	10 (6, 0)	7.5 (6. 1)	7.0 (6, 2)	6.5 (6, 3)	6.0 (6, 4)	5.5 (6, 5)	5.0 (6, 6)	4.5 (6, 7)	4.0 (6, 8)

$f_i^{(0)}(n) < f_i^{(0)}(n-1)$. Hence we arrive at Table III as a first approximation in policy space for $f_i^{(0)}(n)$.

The iteration process can again be described by the equation

$$f_i^{(r)}(n) = \min_{\substack{m \\ j \neq i}} \left[t_{ij}(m) + f_j^{(r-1)}(n-m) \right], \tag{34}$$

TABLE IV

(n) / $f_i^{(1)}(n)$	0	1	2	3	4	5	6	7	8
$f_1^{(1)}(n)$	32 (3, 0)	26 (3, 1)	23.5 (3, 2)	21.5 (3, 2) (3, 3)	19.5 (3, 3) (3, 3)	17.5 (3, 3)	16.5 (2, 2) (3, 4)	15.0 (2, 3)	13.0 (2, 3)
$f_2^{(1)}(n)$	22 (5, 0)	20 (4, 1) (5, 1)	15.5 (5, 1)	14.5 (5, 2)	11.0 (4, 4)	9.0 (4, 4)	8.4 (4, 4)	7.9 (4, 4)	7.5 (4, 4)
$f_3^{(1)}(n)$	17 (4, 0)	15 (4, 0) (4, 1)	13 (4, 1)	12 (4, 2)	11.2 (4, 3)	10.5 (4, 4)	9.9 (4, 5) (5, 5)	9.4 (4, 6) (5, 5)	8.8 (4, 5) (4, 6)
$f_4^{(1)}(n)$	10 (6, 0)	8.0 (6, 1)	7.4 (6, 2)	6.9 (6, 3)	6.5 (6, 4)	6.2 (6, 5)	6.0 (6, 6)	5.8 (6, 7)	5.7 (6, 8)
$f_5^{(1)}(n)$	10 (6, 0)	7.5 (6, 1)	7.0 (6, 2)	6.5 (6, 3)	6.0 (6, 4)	5.5 (6, 5)	5.0 (6, 7)	4.5 (6, 7)	4.0 (6, 8)

$i = 1, 2, \ldots, N-1$; $0 \leqslant m \leqslant n$; with $f_N^{(r)}(0) = 0$. In order to speed up the computation we use tha value of $f_j^{(r)}(n-m)$ on the right-hand side of Eq. (34), whenever it is available instead of $f_f^{(r-1)}(n-m)$. After one iteration we obtain for $f_i^{(1)}(n)$ the values given in Table IV, with the optimal j to go to from point i and amount m of resources used up shown in brackets as (j, m).

Solving again for $f_i^{(2)}(n)$ for $n = 0, 1, 2, \ldots, 8$ and $i = 1, 2, \ldots, 5$ we obtain the same values as $f_i^{(1)}(n)$ showing that it has converged after one iteration to the optimal solution $f_i(n)$. In fact from Table IV one can see that $f_1^{(1)}(8) = 13.0$ (2, 3) giving $t_{12}(3) + t_{24}(4) + t_{46}(1) = 13$ as well as that $f_1^{(1)}(2) = 23.5$ (3, 2) giving $t_{13}(2) + t_{34}(0) + t_{46}(0) = 23.5$.

This method can be easily extended to other network problems for finding the shortest path when a resource has to be allocated.

ACKNOWLEDGMENTS

This research was supported in part by NSERC-Canada Grants A-9247 and A-4051.

REFERENCES

1. R. BELLMAN, "Introduction to the Mathematical Theory of Control Planning," Vol. II, Academic Press, New York, 1971.
2. T. B. BOFFEY, "Graph Theory in Operations Research," The MacMillan & Co., London, 1982.
3. D. DAVIDSON, AND D. J. WHITE, The optimal route problem and the method of approximation in policy space, *J. Math. Anal. Appl.* **32** (1970), 453–455.
4. S. E. DREYFUS AND A. M. LAW, "The Art and Theory of Dynamic Programming,"Academic Press, New York, 1977.
5. R. HARTLEY, "Dynamic Programming in Vector Networks," Notes in Decision Theory Vol. 86 Univ. of Manchester, 1976.
6. E. LAWLER, "Combinatorial Optimization, Methods and Matroids," Holt, Rinehart & Winston, New York, 1979.